Learning Bayesian Tracking for Motion Estimation

Michael Felsberg, Fredrik Larsson

▶ **To cite this version:**

Michael Felsberg, Fredrik Larsson. Learning Bayesian Tracking for Motion Estimation. The 1st International Workshop on Machine Learning for Vision-based Motion Analysis - MLVMA'08, Oct 2008, Marseille, France. inria-00321934

HAL Id: inria-00321934
https://inria.hal.science/inria-00321934

Submitted on 16 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Author manuscript, published in "Dans The 1st International Workshop on Machine Learning for Vision-based Motion Analysis - MLVMA'08 (2008)"

inria-00321934, version 1 - 16 Sep 2008

Learning Bayesian Tracking for Motion Estimation*

Michael Felsberg and Fredrik Larsson

Computer Vision Laboratory, Linköping University, S-58183 Linköping, Sweden

Abstract. A common computer vision problem is to track a physical object through an image sequence. In general, the observations that are made in a single image determine the actual state only partially and information from several views has to be merged. A principled and well-established way of fusing information is the Bayesian framework. In this paper, we propose a novel way of doing Bayesian tracking called *channel-based tracking*. The method is related to grid-based tracking methods, but differs in two aspects: The applied sampling functions, i.e., the bins, are smooth and overlapping and the system and measurement models are learned from a training set. The results from the channel-based tracker are compared to state-of-the-art tracking methods based on particle filters, using a standard dataset from the literature. A simple computer vision experiment is shown to illustrate possible applications.

1 Introduction

Due to the projection into the image plane, the 3D motion of an object cannot be determined uniquely from monocular views. However, if the object motion is non-degenerate, i.e., if the motion is not restricted to certain subspaces, 3D tracking of the object allows one to estimate its full state, i.e., all considered dimensions of the motion state.

1.1 Problem Formulation

Assume a 3D object moving in space and we want to estimate a subspace of its configuration, in practice mostly its 3D position. The observation data consists of consecutive 2D views of the object, taken from a single camera. For simplicity the camera will not move here, but the case where the camera moves is believed to be solvable within the same framework. The relative configuration of image plane and motion trajectory are assumed to be such that all state space dimensions of interest can be estimated, i.e., the system is observable [1]. It is further assumed that the observation data is already extracted from the sequence, i.e., a sequence of observations (image coordinates) is the input. The output of the method

* The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 215078 (DIPLECS) and from the CENIIT project CAIRIS.

inria-00321934, version 1 - 16 Sep 2008

consists of a sequence of 3D coordinates. All models are assumed to be non-linear and the noise terms are assumed as non-Gaussian. Under these circumstances, we believe that only learning systems can provide stable and robust solutions. Therefore, the proposed approach makes *no use of manually engineered system models or measurement models*. Both models are learned from training data, where during the learning phase samples for system state sequences are required. Note that both models are multi-modal and thus non-trivial to learn.

Learning the measurement model implies that *no camera calibration is required* in contrast to state-of-the-art methods that rely on accurate calibration. Coordinate systems can be chosen fairly arbitrarily and according to the requirements of the application in mind. Learning the system model, i.e., learning the dynamics of objects, allows one to change to suitable, application dependent state space coordinate systems. For instance, when it comes to car safety systems one might consider the distance in terms of field of safe travel [2] which is more relevant than the Euclidean distance to the car. In contrast to state-of-the-art methods using predefined system models, the same implementation can be used on different state spaces.

1.2 Related Work

The most relevant work from the literature can be found in the area of non-linear, non-Gaussian Bayesian tracking [3, 4]. In Bayesian tracking, the current state of the system is represented as a probability density function of the system's state space. In the prediction step, this density is modified according to the system model and a new, typically smoother density is obtained as the prediction of the next system state. In the observation step, measurements of the system are used to update the predicted density. Typically, the density is sharpened by inference and it serves as the next system state.

Non-linear, non-Gaussian Bayesian tracking excludes (extended) Kalman filtering. Common approaches are particle filters and grid-based methods [3]. Whereas particle filters apply Monte Carlo methods for approximating the relevant density function, grid based methods discretize the state-space, i.e., apply histogram methods for the approximation. In the case of particle filters, densities are propagated through the models by computing the output for individual particles. Grid-based methods use discretized transition maps to propagate the histograms and are closely related to Bayesian occupancy filtering [5].

An extension to grid based methods is to replace the rectangular histogram bins with overlapping, smooth kernel functions. This leads to the recently proposed *channel-based tracking* [6]. Channel-based tracking implements Bayesian tracking using channel representations [7] and linear mappings on channel representations, so-called associative networks [8]. The term *channel* is well established in vision literature for a representation using a band pass tuning function [9], while some may at times have a reason to view it as a population coding [10, 11]. The main advantage compared to grid-based methods is the reduction of quantization effects and reduced computational effort.

Channel representations can be considered as regularly sampled kernel density estimators [12], implying that channel-based tracking is related to kernel-based prediction for Markov sequences [13, 14]. In the cited work, system models are estimated in a similar way as described here, but the difference is that we consider sampled densities, making the algorithm much faster than the cited work. Another way to represent densities in tracking are Gaussian mixtures (e.g. [15]) and models based on mixtures can be learned using the EM algorithm, cf. [16], although the latter method is restricted to uni-modal cases (Kalman filter).

The main novelty of this paper compared to [6] is the generalization of channel-based tracking to multi-dimensional state vectors, higher-order system models, and generic measurement models. Higher-order system models are required for tracking partially observable states and generic measurement models are obtained through learning. Furthermore, this paper contains a quantitative evaluation of results in comparison to particle filters and grid-based methods.

1.3 Organization of the Paper

The paper is organized as follows. After the introduction, the methods required for further reading are introduced: Bayesian tracking and channel representations of densities. As novelties of this paper, the channel-based tracking algorithm is formulated for the multi-dimensional case and the learning of system and observation models is described. In section 4 channel-based tracking is compared to particle filters and grid-based methods in a standard performance test for Bayesian tracking. A simple vision-based experiment is shown to illustrate applicability in practical problems. In section 5 we discuss the achieved results.

2 Bayesian Tracking and Channel Representations

Channel-based tracking can be considered as a generalization of grid-based methods for implementing non-linear, non-Gaussian Bayesian tracking. Hence we give a brief overview on Bayesian tracking and channel representations.

2.1 Bayesian Tracking

For the introduction of concepts from Bayesian tracking we adopt the notation from [3]. Bayesian tracking is commonly defined in terms of a process model $\mathbf{f}$ and a measurement model $\mathbf{h}$, distorted by i.i.d. noise $\mathbf{v}$ and $\mathbf{n}$

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \tag{1}$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \ . \tag{2}$$

The symbol $\mathbf{x}_k$ denotes the system state at time k and $\mathbf{z}_k$ denotes the observation made at time k. Both models are in general non-linear and time-dependent.

The current state can be estimated, given that the previous state and all previous observations are known, by using the prediction equation. Assuming a

inria-00321934, version 1 - 16 Sep 2008

Markov process of order one allows us to consider the conditional density of the novel state as an integral over its conditional density given the previous state

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1} \ . \tag{3}$$

Taking into account the new measurement, the prediction is updated through

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})\, d\mathbf{x}_k} \ . \tag{4}$$

In the case of non-linear problems with multi-modal densities, basically two relevant approaches for implementing (3) and (4) are known: The particle filter and grid-based methods. Particle filters are considered superior to grid-based methods concerning accuracy [3].

Particle filter methods apply Monte Carlo simulation for approximating the relevant densities, and the crucial step is the re-sampling of particles from the previous estimates in order to avoid degeneracy and loss of diversity. Cumulative histograms are required for the re-sampling step [4], and in case of many particles and high-dimensional spaces, the method suffers from the computational burden.

Grid-based methods assume a discrete state space such that the densities are approximated with histograms. Thus, conditional probabilities of state transitions are replaced with linear mappings. In contrast to [3] where densities were formulated using Dirac distributions weighted with discrete probabilities, we assume band-limited densities and apply sampling theory. However, the final equations for grid-based tracking and the model assumptions remain the same.

For the remainder of this section, sampling is meant in the signal processing sense and not in its meaning as a stochastic sampling. Sampling the posterior of the previous time step gives us

$$w^i_{k-1|k-1} \triangleq p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) * \delta(\mathbf{x}^i - \mathbf{x}_{k-1}) \tag{5}$$

where $*$ denotes convolution and $\delta(\mathbf{x}^i - \mathbf{x})$ is the Dirac impulse at $\mathbf{x}^i$. Note that the sampling itself is time independent. Accordingly, the sampled prior pdf at time k and the sampled posterior at time k are obtained as

$$w^i_{k|k-1} \triangleq p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) * \delta(\mathbf{x}^i - \mathbf{x}_k) \tag{6}$$

$$w^i_{k|k} \triangleq p(\mathbf{x}_k|\mathbf{z}_{1:k}) * \delta(\mathbf{x}^i - \mathbf{x}_k) \ . \tag{7}$$

Plugging (5) and (3) into (6) and applying the power theorem gives us

$$w^i_{k|k-1} = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1} * \delta(\mathbf{x}^i - \mathbf{x}_k) = \sum_j f^{ij}_k w^j_{k-1|k-1} \tag{8}$$

$$\text{where } f^{ij}_k = p(\mathbf{x}_k|\mathbf{x}_{k-1}) * \delta(\mathbf{x}^i - \mathbf{x}_k) * \delta(\mathbf{x}^j - \mathbf{x}_{k-1}). \tag{9}$$

Accordingly, plugging (6) and (4) into (7) gives us

$$w^i_{k|k} = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) * \delta(\mathbf{x}^i - \mathbf{x}_k)}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})\, d\mathbf{x}_k} = \frac{h^i_k(\mathbf{z}_k)w^i_{k|k-1}}{\sum_j h^j_k(\mathbf{z}_k)w^j_{k|k-1}} \tag{10}$$

$$\text{where } h^i_k(\mathbf{z}_k) = p(\mathbf{z}_k|\mathbf{x}_k) * \delta(\mathbf{x}^i - \mathbf{x}_k). \tag{11}$$

inria-00321934, version 1 - 16 Sep 2008

inria-00321934, version 1 - 16 Sep 2008

Note that (8) and (10) are exactly the same equations as in the formulation of grid-based methods in [3], except for change of notation. Re-writing grid-based methods in terms of the sampling theorem allows us to analyze approximation errors more systematically. Grid-based methods require the more samples the higher the upper band limit of the pdf, i.e., the wider the characteristic function $\varphi_{\mathbf{x}}(\mathbf{t}) = E\{\exp(i\mathbf{t}^T\mathbf{x})\}$. Furthermore, the sampling-based formulation makes it easier to switch to other sampling schemes than the ordinary impulse sampling.

2.2 Channel Representations of Densities

The channel representation [7, 17] can be considered as a way of sampling continuous functions or, alternatively, as histograms where the bins are replaced with smooth, overlapping basis functions, see e.g. [18].

Consider a density function $p(\mathbf{x})$ as a continuous signal that is sampled with a smooth basis function, e.g., a B-spline. It is important to realize here that the sampling takes place in the dimensions of the stochastic variables, not along the time axis k. Sampling is meant in the signal processing sense, not in its meaning as a stochastic sampling. It has been shown in the literature that an averaging of a stochastic variable in channel representation is equivalent to the sampled kernel density estimator with the channel function as kernel function [12]. For the purpose of Bayesian tracking, we replace the sampled densities (5) – (7) with

$$w^i_{k_1|k_2} \triangleq p(\mathbf{x}_{k_1}|\mathbf{z}_{1:k_2}) * b(\mathbf{x}^i - \mathbf{x}_{k_1}) \tag{12}$$

where $b(\mathbf{x})$ is a channel basis function. For the remainder of this paper it is chosen as [19]

$$b(\mathbf{x}) \triangleq \frac{2a}{\pi} \begin{cases} \prod_n \cos^2(ax_n) & |x_n| < \frac{\pi}{2a} \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Here a determines the relative width, i.e., the sampling density. According to [12], the channel representation reduces the quantization effect compared to ordinary histograms by a factor of up to 20. Switching from histograms to channels thus allows us to reduce computational load by using fewer bins, to increase the accuracy for the same number of bins, or both at the same time. In theory, the reduction of bins is limited by the upper band limit of the density function, but in practice the number of drawn samples (in a stochastic sense now) is often much too low anyway and regularization becomes necessary.

For performing maximum likelihood or MAP estimation using channels, a suitable algorithm for extracting the maximum of the represented distribution is required. For $\cos^2$-channels with a spacing of $\frac{\pi}{3a}$, an optimal algorithm in least-squares sense is obtained in the one-dimensional case as [19]

$$\hat{x}_{k_1} = l + \frac{1}{2a} \arg\left[\sum_{j=l}^{l+2} w^j_{k_1|k_1} \exp(i2a(j-l))\right]. \tag{14}$$

N-dimensional decoding is obtained by local marginalization in a window of size 3^N and subsequent decoding of the N marginals. The index l of the decoding window is chosen using the maximum sum of a consecutive triplet of coefficients.

inria-00321934, version 1 - 16 Sep 2008

3 Channel-Based Tracking

This section contains the multi-dimensional channel-based tracking algorithm and the learning method for the system and the measurement models.

3.1 Channel-Based Tracking Algorithm

When plugging channel representations into (5) – (7), the two conditional densities $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{z}_k|\mathbf{x}_k)$ have to be considered more closely. The power theorem which has been used to derive (8) and (10) does not hold if we sample with channels instead of impulses, because some high-frequency content is removed and thus the scalar product between sampled densities will always be less than the integral product of the corresponding continuous densities.

However, if the densities are band-limited from the start, the regularization by the channel basis functions removes no or only little high-frequency content. Hence, the power theorem holds approximately, i.e., the difference between the scalar product and the integral product can be neglected. This has been confirmed in experiments with synthetically generated densities. Henceforth, (8) and (10) can be applied for the channel-based density representations as well.

For what follows, the coefficients of (8) are summarized in the matrix $\mathbf{F}_k = \{f_k^{ij}\}$ and the coefficients of (10) are summarized in the vector-valued function $\mathbf{h}_k(\mathbf{z}_k) = \{h_k^j(\mathbf{z}_k)\}$. In the following section, both operators will be learned from a set of training data. This requires that both remain stationary and we remove the time index k (not from $\mathbf{z}_k$ though): $\mathbf{F}$ and $\mathbf{h}(\mathbf{z}_k)$. This removes absolute time reference from the model. The posterior density is now obtained by

$$\mathbf{w}_{k|k-1} = \mathbf{F}\mathbf{w}_{k-1|k-1} \tag{15}$$

$$\mathbf{w}_{k|k} = \frac{\mathbf{h}(\mathbf{z}_k) \cdot \mathbf{w}_{k|k-1}}{\mathbf{h}^T(\mathbf{z}_k)\mathbf{w}_{k|k-1}} \ , \tag{16}$$

where $\cdot$ is the element-wise product, i.e., the enumerator remains a vector.

We have not yet addressed the state space and the prediction and measurement model. The models will be treated separately in the subsequent section, for the moment it is assumed that they are known. The state space should contain sufficiently many degrees of freedom to describe the observed phenomena. Typical choices are first or second order Euclidean motion states, since the Markov model of order one applied in (3) only keeps track of the previous state.

The choice of such a complex state space requires prior knowledge about the problem which is not available in the considered task, as mentioned in Sect. 1.1. Applying the Markov theorem the other way round, an equivalent setting is to use only positions as states and to apply a higher-order Markov model. Hence, more than just the previous state is considered in the prediction step:

$$\mathbf{w}_{k|k-1,k-2,\ldots,k-n} = [\mathbf{F}_1\mathbf{F}_2\ldots\mathbf{F}_n][\mathbf{w}_{k-1|k-1}^T\mathbf{w}_{k-2|k-2}^T\cdots\mathbf{w}_{k-n|k-n}^T]^T \tag{17}$$

In the observation step (16) the prior pdf has to be changed accordingly.

inria-00321934, version 1 - 16 Sep 2008

In (17), the state space is represented by the *concatenation* of channel representations. Using concatenation of channels instead of an outer product of channel vectors leads to *linear asymptotic growth* of computational complexity instead of exponential growth. Using a concatenation of channel vectors corresponds to a marginalization of the joint density of previous states. In theory this could lead to predictions based on non-corresponding previous states, but following the line of arguments in [20], this is very unlikely to happen for channel based linear mappings. The whole channel-based tracking algorithm is summarized in Algorithm 1. The mentioned prior is treated in the following section.

3.2 Learning of System and Measurement Models

A particular feature of channel-based tracking is that the system model $\mathbf{f}$ and the measurement model $\mathbf{h}$ can easily be learned - which is different from most particle-filter based methods which need pre-specified models. The system model is trained from a sequence of states, i.e., at some time the system needs access to the entire state space. This can be thought of as a calibration phase in the case of computer vision applications or as a bootstrapping phase in the case of cognitive robotics. Another option is to observe the entire state space of another, accessible system, e.g., observing the own car, and using the model to predict an inaccessible system, e.g., another car.

In all subsequent experiments, the system model is trained by estimating the matrix $[\mathbf{F}_1\mathbf{F}_2\dots\mathbf{F}_n]$ from the covariance of the state channel vector at time k and the n previous instances. Since the model matrix corresponds to the conditional pdf and not to the joint pdf, the covariance is normalized with the marginal distribution for the n previous states (see also [14], plugging (3.3) into (2.7))

$$[\hat{\mathbf{F}}_1\hat{\mathbf{F}}_2\dots\hat{\mathbf{F}}_n] = \frac{\sum_{k=n}^{K_{\max}} \mathbf{w}_{k|k}[\mathbf{w}_{k-1|k-1}^T\mathbf{w}_{k-2|k-2}^T\cdots\mathbf{w}_{k-n|k-n}^T]}{\mathbf{1}\sum_{k=n}^{K_{\max}}[\mathbf{w}_{k-1|k-1}^T\mathbf{w}_{k-2|k-2}^T\cdots\mathbf{w}_{k-n|k-n}^T]} \tag{18}$$

where $\mathbf{1}$ denotes a one-vector of suitable size and the quotient is evaluated pointwise. An important advantage of the covariance-based method for estimating the model matrix is that one can easily incrementally update the matrix by adding the covariance of the updated new state and the n previous states to the enumerator and the marginal to the denominator.

Algorithm 1 channel-based tracking algorithm.

Require: $[\mathbf{F}_1\mathbf{F}_2\dots\mathbf{F}_n]$, $\mathbf{h}(\mathbf{z}_k)$ and prior are known
1: set $[\mathbf{w}_{n-1|n-1}^T\mathbf{w}_{n-2|n-2}^T\dots\mathbf{w}_{0|0}^T]$ according to prior
2: **for** $k = n$ to $K_{\max}$ **do**
3: compute $\mathbf{w}_{k|k-1,k-2,\dots,k-n}$ according to (17)
4: acquire $\mathbf{z}_k$
5: compute $\mathbf{w}_{k|k}$ according to (16)
6: compute $\hat{\mathbf{x}}_k$ by applying (14) to $\mathbf{w}_{k|k}$
7: **end for**

inria-00321934, version 1 - 16 Sep 2008

For the first n steps, no complete previous state sequence is available and the prediction equation cannot be computed using the model matrix above. Instead, the empirical distribution for the first n states is stored and used as a prior in line 1 of Algorithm 1.

The measurement model is the most difficult part in the processing chain. Since no analytic formulations of the measurement equation are available, $\mathbf{h}(\mathbf{z}_k)$ cannot be a continuous function in $\mathbf{z}_k$ and has to be approximated by a suitable scheme. For the sake of simplicity, the observation data is also represented in a channel representation such that the measurement model becomes a product of the current observation channel vector $\mathbf{v}_k$ and a matrix $\mathbf{H}$

$$\mathbf{h}(\mathbf{z}_k) \approx \text{recode}(\mathbf{v}_k^T \mathbf{H}). \tag{19}$$

Here, recode$(\cdot)$ denotes the subsequent decoding (14) of modes, pruning, and weighted re-encoding of modes, which corresponds to an inhibition of side-maxima. The matrix $\mathbf{H}$ is estimated in a similar way as the system model

$$\hat{\mathbf{H}} = \frac{\sum_{k=1}^{K_{\max}} \mathbf{v}_k \mathbf{w}_{k|k}^T}{\mathbf{1} \sum_{k=1}^{K_{\max}} \mathbf{w}_{k|k}^T} \tag{20}$$

4 Experiments

In this section, experiments validating the concept of channel-based Bayesian tracking are discussed. In Sect. 4.1 we evaluate our presented method on the classical Carlin's experiment. We compare our result to state-of-the-art methods such as the SIR particle filter and the likelihood particle filter and show that we can achieve competitive results. We demonstrate the validity of our method on a real world visual experiment in Sect. 4.2.

4.1 Carlin's Experiment

In the first experiment, the following system is considered

$$\begin{aligned} x_k &= \tfrac{x_{k-1}}{2} + \tfrac{25 x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + v_{k-1} \\ z_k &= \tfrac{x_k^2}{20} + n_k \end{aligned} \tag{21}$$

where v_{k-1} and n_k is zero mean Gaussian white noise with variances 10.0 and 1.0 respectively. This system is highly nonlinear regarding both the system and measurement equation and the symmetric nature of the measurement equation poses a challenging task. This example has been used for evaluation in several publications e.g. [21, 22, 3]. We use the Root Mean Squared Error (RMSE) as a measure of performance to be able to compare our result to the ones reported in [3]. In our setup the initial state $\mathbf{x}_0$ is set to 8 according to $x_k = 0$ for $k < 0$.

The true state and the estimated result for one evaluation can be seen in Fig. 1. Note that for most of the time the results are remarkable good. However

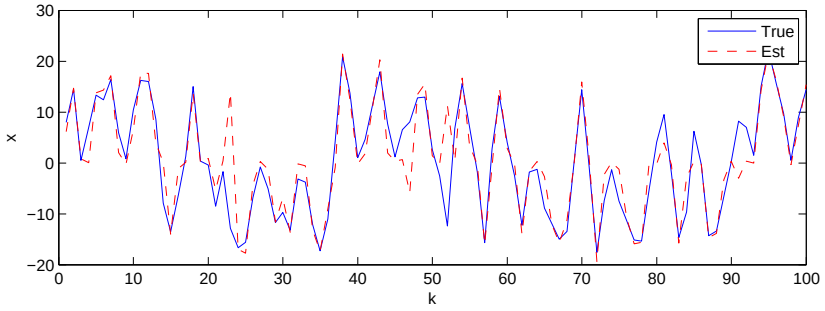

Fig. 1. The true (solid line) and estimated state (dashed line) for the Carlin's experiment.

for a few instances, e.g. $k = 23$, the estimated state seems to be mirrored in $x = 0$. This can be explained by the measurement function. From the measurement function alone, there is no way to tell if we are in $-x$ or $+x$.

A comparison with the results in [3] is shown in table 1. The RMSE presented for our method was obtained with a second order model using 12 $\cos^2$-channels for both the observation and state space. We trained our method on 100 sets and performed 100 evaluation runs. It should be noted that we did model the additive time dependent component of the system, i.e. the $\cos(1.2k)$ term, since it does not comply with the stationary assumption needed for learning. However, we did learn the remaining components of the system model as well as the full observation model. The likelihood particle filter performs best. It should be noted that the test scenario is heavily biased toward the particle filters, since the only unknown components for the particle filters are the different noise components, while the system and the measurement model are fully known, which in a real world scenario must be considered unrealistic due to model errors. Our results incorporate system model and measurement model errors as well as noise errors. Despite these facts, we still manage to produce competitive result.

Table 1. RMSE obtained on Carlin's experiment. Our method is third in performance even though we do not model the system or measurement model. All results except for CBT (12 channels) were taken from [3] (50 particles and 50 grid points).

Algorithm	RMSE
Extended Kalman filter	23.19
Approximate grid-Based Filter	6.09
Regularized Particle filter	5.55
SIR Particle filter	5.54
Channel Based Tracking	**5.43**
Auxiliary Particle filter	5.35
Likelihood Particle filter	5.30

inria-00321934, version 1 - 16 Sep 2008

inria-00321934, version 1 - 16 Sep 2008

4.2 Computer Vision Experiment

The scenario consists of a spherical object, an orange, attached to the roof by a string. We captured images of the object by an uncalibrated stereo camera setup where external and internal parameters are unknown and lens distortions are not compensated.[1] The observations $\mathbf{z}_k$ are the 2D position of the object in the right image and the states $\mathbf{x}_k$ are the position in the left image. We used a third order model with 17 $\cos^2$-channels for both the observation and state space and used 37 frames for training. The resulting $\mathbf{F}$ matrix is visualized in Fig. 3. At frame 38 we simulate a sensor failure and from here on rely on channel-based tracking. The tracking result 14-16 frames after the sensor failure can be seen in Fig. 2. The entire sequence is available at http://www.diplecs.eu/publications.

[1] Actually, offline data has been used consisting of two image sequences only. One of the sequences contained two frame-drops, which were automatically discovered by the channel-based tracking.

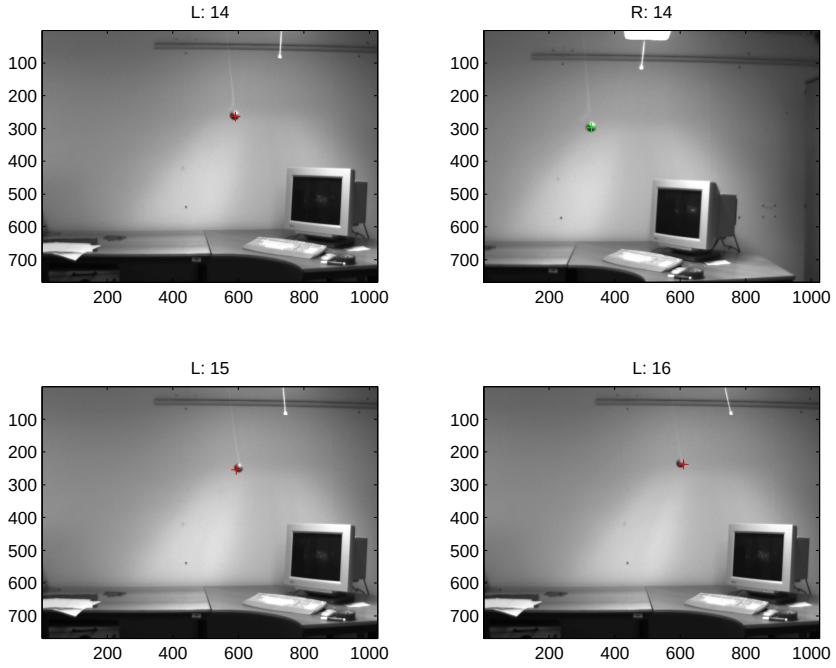

Fig. 2. The tracking result 14, 15 and 16 frames after the simulated sensor failure. The state (i.e. the position in the left image) is obtained by channel-based tracking given the measurements (i.e. the position in the right image). The cross in the upper left image is the estimated position at 14 frames after the simulated sensor failure given the measurement in the corresponding right image, upper right. The lower images show the estimated position after 15 and 16 frames.

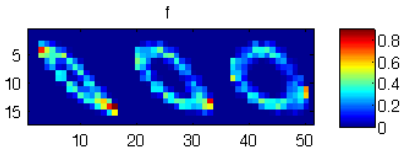

Fig. 3. Visualization of the **F**-matrix that has been learned during the first 37 frames. Note the similarity to Lissajous figures of order (1,1).

Table 2. Comparison to cited work. CBT refers to channel-based tracking and [3] to the four particle filter methods from table 1.

Method	[13]	[14]	[15]	[16]	[3]	**CBT**
learned models	system	system	no	system	no	**both**
multi-modal densities	no	yes	yes	no	yes	**yes**
fast implementation	no[2]	no[2]	no[3]	yes	yes	**yes**

[2] The continuous formulation requires re-evaluation of all kernels for each new sample.
[3] Nothing is said in the paper about computational complexity / performance.

5 Conclusion

In this paper, a novel variant of Bayesian tracking has been proposed: channel-based tracking with learned models. The approach is related to grid-based methods, but uses smooth, overlapping bins and the system and measurement models are acquired through learning. A number of advantages have been postulated and a standard experiment and a vision experiment have been presented.

The most important *advantage of channel-based tracking* compared to particle filters is that competitive results are achieved while the *system model and measurement model are learned from given state and observation data*. In Carlin's experiment, channel-based tracking performs similarly well as particle filters and significantly better than grid-based methods, both concerning the number of bins and the accuracy. However, the accuracy is slightly lower than for state-of-the-art particle filter methods, which had to be expected, since the particle filter methods make use of analytic system and measurement models.

In a second, qualitative computer vision experiment it has been shown that channel-based tracking can be applied to *learn the mapping from uncalibrated cameras to dynamic object states*. This type of experiment can be considered as a prototype for estimation problems under partial occlusion or sensor failure. Other properties in relation to cited work is summarized in table 2.

References

1. Dahl, O., Nyberg, F., Heyden, A.: On observer error linearization for perspective dynamic systems. In: American Control Conference. (2007) 266–268

inria-00321934, version 1 - 16 Sep 2008

2. Gibson, J.J., Crooks, L.E.: A theoretical field-analysis of automobile-driving. The American Journal of Psychology **L1**(3) (1938)
3. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing **50**(2) (2002) 174–188
4. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. International Journal of Computer Vision **29**(1) (1998) 5–28
5. Coué, C., Fraichard, T., Bessière, P., Mazer, E.: Using Bayesian programming for multi-sensor multitarget tracking in automotive applications. In: International Conference on Robotics and Automation. (2003)
6. Felsberg, M., Granlund, G.: Fusing dynamic percepts and symbols in cognitive systems. In: International Conference on Cognitive Systems. (2008)
7. Granlund, G.H.: An Associative Perception-Action Structure Using a Localized Space Variant Information Representation. In: Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC), Kiel, Germany (September 2000)
8. Johansson, B., Elfving, T., Kozlov, V., Censor, Y., Forssén, P.E., Granlund, G.: The application of an oblique-projected landweber method to a model of supervised learning. Mathematical and Computer Modelling **43** (2006) 892–909
9. Howard, I.P., Rogers, B.J.: Binocular Vision and Stereopsis. Oxford University Press, Oxford, UK (1995)
10. Zemel, R.S., Dayan, P., Pouget, A.: Probabilistic interpretation of population codes. Neural Computation **10**(2) (1998) 403–430
11. Pouget, A., Dayan, P., Zemel, R.: Information processing with population codes. Nature Reviews – Neuroscience **1** (2000) 125–132
12. Felsberg, M., Forssén, P.E., Scharr, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(2) (2006) 209–222
13. Georgiev, A.A.: Nonparamtetric system identification by kernel methods. IEEE Trans. on Automatic Control **29**(4) (1984)
14. Yakowitz, S.J.: Nonparametric density estimation, prediction, and regression for markov sequences. Journal of the American Statistical Association **80**(389) (1985)
15. Han, B., Joo, S.W., Davis, L.S.: Probabilistic fusion tracking using mixture kernel-based Bayesian filtering. In: IEEE Int. Conf. on Computer Vision. (2007)
16. North, B., Blake, A.: Learning dynamical models using expectation-maximisation. In: IEEE Int. Conf. on Computer Vision. (1998)
17. Snippe, H.P., Koenderink, J.J.: Discrimination thresholds for channel-coded systems. Biological Cybernetics **66** (1992) 543–551
18. Pampalk, E., Rauber, A., Merkl, D.: Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In: Proceedings of the International Conference on Artifical Neural Networks (ICANN'02), Madrid, Spain, Springer (August 27-30 2002) 871–876
19. Forssén, P.E.: Low and Medium Level Vision using Channel Representations. PhD thesis, Linköping University, Sweden (2004)
20. Jonsson, E., Felsberg, M.: Correspondence-free associative learning. In: International Conference on Pattern Recognition, Hong Kong (August 2006)
21. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Radar and Signal Processing, IEE Proceedings F **140**(2) (Apr 1993) 107–113
22. Carlin, B.P., Polson, N.G., Stoffer, D.S.: A Monte Carlo approach to nonnormal and nonlinear state-space modeling. Journal of the American Statistical Association **87**(418) (1992) 493–500

inria-00321934, version 1 - 16 Sep 2008